ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE CM2
OBJECT MODULE PLACED IN CM2.OBJ
COMPILER INVOKED BY: :F0: CM2.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982) XREF

```
                $title ('codemacro module 2')
  1             cm2:
                do;

                /*

                  modified  3/26/81  R. Silberstein

                */

                /*    This is the module to build new instructions
                      which is not present in the already existing
                      system. */

                $include (:f1:macro.lit)
              =  $nolist
                $include (:f1:equals.lit)
              =  $nolist
                $include (:f1:struc.lit)
              =  $nolist
                $include (:f1:cmacd.lit)
              =  $nolist
                $include (:f1:cm.lit)
              =  $nolist
                $include (:f1:symb.ext)
              =  $nolist
                $include (:f1:subr1.ext)
              =  $nolist
                $include (:f1:subr2.ext)
              =  $nolist
                $include (:f1:scan.ext)
              =  $nolist
                $include (:f1:files.ext)
              =  $nolist
                $include (:f1:exglob.ext)
              =  $nolist
```

CP/M-86 v.1.1 (vol. IV)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
$eject

/*  D E C L A R A T I O N  F O R  'C O D E M A C R O'  P A R T

    Extended version of ASM86 */

148  1    dcl   cm$name(80)       byte                    ,
             codemacro$found     byte                    ,
             cm$n$pt             address                 ,
             cm$n$var            based     cm$n$pt       byte,
             cm$n$lg             byte                    ,
             par$name(30)        byte                    ,
             pmpt                address                 ,
             cm$pm$var           based     pmpt          byte,
             cm$error            byte      public        ,
             cm$body(100)        byte                    ,
             cmpt                address   public        ,
             cm$b$var            based     cmpt          byte,
             cm$w$var            based     cmpt          addr,
             cm$counter          addr                    ,
             global$cm$error     byte      public        ,
             cm$body$full        byte                    ,
             ant$par             byte                    ,
             cm$list$overflow    byte      public;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
                $eject

                /*  VARIOUS SUBROUTINES */

149   1         legal$parameter: PROC(lg,ptr,ptr2) byte public;
150   2           dcl (lg,i) byte,
                      (ptr,ptr2) address,
                       no based ptr2 byte;
151   2           i=0;
152   2           do no=0 to cm$body(2)-1;
153   3             if parname(i+lg) = 0 and
                    equal(lg,ptr,.par$name(i)) then return true;
155   3             do while par$name(i:=i+1) <> 0;
156   4             end$while;
157   3             i=i+1;
158   3           end;
159   2           return false;
160   2         end$proc legal$parameter;

161   1         legal$spec$letter: PROC(l) byte;
162   2           dcl (l,i) byte;
163   2           dcl table(8) byte data ('ACDEMRSX');
164   2           do i=0 to last(table);
165   3             if table(i)=l then return i;
167   3           end;
168   2           return i;
169   2         end$proc legal$spec$letter;

170   1         legal$mod$letter: PROC(l) byte;
171   2           dcl (l,i) byte;
172   2           dcl table(4) byte data ('BWDS');
173   2           do i=0 to last(table);
174   3             if table(i) = l and i < 3 then return i;
176   3             if table(i) = l and i = 3 then$do
178   4               if accum(2) = 'B' then return 3;
                    end$if;
181   3           end;
182   2           return i;
183   2         end$proc legal$mod$letter;

184   1         legal$register: PROC byte;
185   2           declare disp byte;
186   2           if token.type <> reg then return 0;
188   2           disp=0;
189   2           if token.descr=byt then disp=8;
191   2           if token.descr=dwrd then disp=16;
193   2           return token.value + disp + 1;
194   2         end$proc legal$register;

195   1         legal$seg$reg: PROC byte public;
196   2           if token.type <> reg then return 0;
198   2           if token.descr <> dwrd then return 0;
200   2           return token.value + 1;
201   2         end$proc legal$seg$reg;

202   1         put$b: PROC(b) public;
203   2           dcl b byte;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
 204  2        cm$counter=cm$counter+1;
 205  2        if cm$counter > 99 then$do
 207  3          cm$error=true;
 208  3          cm$body$full=true;
 209  3          return;
 210  3        end$if;
 211  2        cm$b$var=b;
 212  2        cmpt=cmpt+1;
 213  2      end$proc put$b;

 214  1      put$w: PROC(w) public;
 215  2        dcl w addr;
 216  2        cm$counter=cm$counter+2;
 217  2        if cm$counter > 99 then$do
 219  3          cm$error=true;
 220  3          cm$body$full=true;
 221  3          return;
 222  3        end$if;
 223  2        cm$w$var=w;
 224  2        cmpt=cmpt+2;
 225  2      end$proc put$w;

 226  1      update$cm$lists: PROC byte;
 227  2        dcl listptr address,next based listptr address;
 228  2        dcl ptr address;
 229  2        ptr=.listptr;
 230  2        if findcodemacro(cm$n$lg,.cm$name(0),ptr)
                then$do
 232  3          do while next <> 0;
 233  4            listptr=next;
 234  4          end$while;
 235  3        else$do
 237  3          if not new$cm$name(cm$n$lg,.cm$name(0),ptr)
                  then return false;
              end$if;
 240  2        next=freept;
 241  2        if not new$cm$body(cm$counter,.cm$body(0))
                then return false;
 243  2        return true;
 244  2      end$proc update$cm$lists;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
                $eject

                /* level 4 in the syntax-tree of coeemacro building */

245   1         register$rout: PROC (1);
246   2           dcl 1 byte;
247   2           call put$b(1);
248   2         end$proc register$rout;

249   1         range$rout: PROC;

250   2           put$range: PROC(time);
251   3             dcl time byte;
252   3             if token.type = number then$do
254   4               if time = first then$do
256   5                 cm$b$var=cm$b$var+numberrange;
257   5                 cmpt=cmpt+1;
258   5               end$if;
259   4               if token.value > 0ffh then$do
261   5                 cm$error=true; /* too large number */
262   5                 return;
263   5               else$do
265   5                 call put$b(token.value);
266   5               end$if;
267   4             else$do
269   4               if time = first then$do
271   5                 cm$b$var=cm$b$var+registerrange;
272   5                 cmpt=cmpt+1;
273   5               end$if;
274   4               do case legal$register;
275   5                 do; /* error, expecting a register spec. */
276   6                   cm$error=true;
277   6                   return;
278   6                 end;
279   5                 call register$rout(rax);
280   5                 call register$rout(rcx);
281   5                 call register$rout(rdx);
282   5                 call register$rout(rbx);
283   5                 call register$rout(rsp);
284   5                 call register$rout(rbp);
285   5                 call register$rout(rsi);
286   5                 call register$rout(rdi);
287   5                 call register$rout(ral);
288   5                 call register$rout(rcl);
289   5                 call register$rout(rdl);
290   5                 call register$rout(rbl);
291   5                 call register$rout(rah);
292   5                 call register$rout(rch);
293   5                 call register$rout(rdh);
294   5                 call register$rout(rbh);
295   5                 call register$rout(res);
296   5                 call register$rout(rcs);
297   5                 call register$rout(rss);
298   5                 call register$rout(rds);
299   5               end$case;
300   4             end$if;
301   3             call scan;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
302  3      end$proc put$range;


303  2      s$range: PROC;
304  3        cmpt=cmpt-1;
305  3        cm$b$var=cm$b$var+singlerange;
306  3        call put$range(first);
307  3      end$proc s$range;


308  2      d$range: PROC;
309  3        cmpt=cmpt-1;
310  3        cm$b$var=cm$b$var+doublerange;
311  3        call put$range(first);
312  3        if accum(0) <> comma then$do
314  4          cm$error=true;
315  4          return;
316  4        end$if;
317  3        call scan;
318  3        call put$range(second);
319  3        if accum(0) <> rightpar then$do
321  4          cm$error=true;
322  4          return;
323  4        end$if;
           end$proc d$range;

         /* mainpart of range routine */

325  2        call scan; /* skip left paranthesis */
326  2        if nextch = comma then call d$range;
328  2                          else call s$range;
329  2        call scan;
330  2      end$proc range$rout;

331  1      spec$letter$rout: PROC(1);
332  2        dcl l byte;
333  2        call put$b(l);
334  2      end$proc spec$letter$rout;

335  1      mod$letter$rout: PROC(1);
336  2        dcl l byte;
337  2        call put$b(l);
338  2      end$proc mod$letter$rout;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
                $eject

                /* level 3 in the syntax-tree of codemacro building */

339   1         par$descr$rout: PROC;
340   2           call copy(acclen,.accum(0),.cm$pm$var);
341   2           pmpt=pmpt+acclen;
342   2           cm$pm$var=0; /* end of par.name */
343   2           pmpt=pmpt+1;
344   2           call scan;
345   2           if accum(0) <> colon then$do
                    /* error, expected colon in parameterdecl */
347   3             cm$error=true;
348   3           end$if;
349   2           call scan;
350   2           do case legal$spec$letter(accum(0));
351   3             call spec$letter$rout(speca);
352   3             call spec$letter$rout(specc);
353   3             call spec$letter$rout(specd);
354   3             call spec$letter$rout(spece);
355   3             call spec$letter$rout(specm);
356   3             call spec$letter$rout(specr);
357   3             call spec$letter$rout(specs);
358   3             call spec$letter$rout(specx);
359   3             do; /* error, expected specifier letter */
360   4               cm$error=true;
361   4               return;
362   4             end;
363   3           end$case;

364   2           do case legal$mod$letter(accum(1));
365   3             call mod$letter$rout(modb);
366   3             call mod$letter$rout(modw);
367   3             call mod$letter$rout(modd);
368   3             call mod$letter$rout(modsb);
369   3             call mod$letter$rout(nomod); /* no modletter */
370   3           end$case;
371   2           call scan;
372   2           if accum(0) = leftpar then call range$rout;
374   2           cm$body(2)=cm$body(2)+1;
375   2         end$proc par$descr$rout;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
            $eject

            /* level 2 in the syntax-tree of codemacro building */

            /* Procedure to initialize temporary storage and pointers
               conserning the building of codemacro */
376   1     init$cm$rout: PROC public;
377   2       cm$n$pt=.cm$name(0);
378   2       cmpt=.cm$body(0)+3; /* correcting fo  the head */
379   2       pmpt=.par$name(0);
380   2       CALL FILL (0, LENGTH (CM$NAME), .CM$NAME);
381   2       CALL FILL (0, LENGTH (CM$BODY), .CM$BODY);
382   2       CALL FILL (0, LENGTH (PAR$NAME), .PAR$NAME);
383   2       ant$par=0;
384   2       cm$counter=3;
385   2       cm$body$full=false;
386   2       cm$list$overflow=false;
387   2     end$proc init$cm$rout;

388   1     name$rout: PROC byte public;
389   2       if token.type <> ident then return false;
391   2       call copy(acclen,.accum(0),.cm$name(0));
392   2       cm$n$lg=acclen;
393   2       call scan;
394   2       return true;
395   2     end$proc name$rout;

396   1     formal$list$rout: PROC public;
397   2       do while token.type = ident;
398   3         call par$descr$rout;
399   3         if accum(0) <> ',' then return;  /* end of parameters */
401   3         call scan;
402   3       end$while;
403   2     end$proc formal$list$rout;

404   1     terminate$cm$rout: PROC public;
405   2       if global$cm$error then$do
                /* error present in codemacrodef, */
                /* no updating of codemacrolist   */
407   3         return;
408   3       end$if;
409   2       if not update$cm$lists
                then$do
411   3         cm$error=true; /* overflow, no more vacant memory */
412   3         cm$list$overflow=true;
413   3         return;
414   3       end$if;
            end$proc terminate$cm$rout;


416   1     end$module cm2;
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
147	0000H	1	ACCLEN . . . . . BYTE EXTERNAL(59) 34 341 391 392
147	0000H	80	ACCUM. . . . . BYTE ARRAY(80) EXTERNAL(60) 178 312 319 340 345 350 364 372 391 399
2			ADDR . . . . . LITERALLY 18 19 25 28 31 34 37 40 43 46 49 52 55 58 64 82 85 88 91 143 147 148 215
72	0000H		ALPHANUMERIC . . . PROCEDURE BYTE EXTERNAL(20) STACK=0000H
148	0115H	1	ANTPAR . . . . . BYTE 383
75	0000H		ASCIICHAR. . . . . PROCEDURE BYTE EXTERNAL(21) STACK=0000H
45	0000H	2	ASCIIPTR . . . . . WORD PARAMETER 46
202	0004H	1	B. . . . . BYTE PARAMETER AUTOMATIC 203 211
147	0008H	1	BASEINDEX. . . . . BYTE MEMBER(OPERANDS)
8			BASEREGBIT . . . . LITERALLY
8			BASEREGCOUNT . . . LITERALLY
8			BREGBIT. . . . . LITERALLY
8			BREGCOUNT. . . . . LITERALLY
4			BYT. . . . . LITERALLY 189
69	0000H	1	CH . . . . . BYTE PARAMETER 70
72	0000H	1	CH . . . . . BYTE PARAMETER 73
75	0000H	1	CH . . . . . BYTE PARAMETER 76
63	0000H	1	CH . . . . . BYTE PARAMETER 64
110	0000H	1	CH . . . . . BYTE PARAMETER 111
78	0000H	1	CH . . . . . BYTE PARAMETER 79
113	0000H	1	CH . . . . . BYTE PARAMETER 114
66	0000H	1	CH . . . . . BYTE PARAMETER 67
107	0000H	1	CH . . . . . BYTE PARAMETER 108
147	0000H	2	CIP. . . . . WORD EXTERNAL(51)
136	0000H		CLOSEHEX . . . . . PROCEDURE EXTERNAL(46) STACK=0000H
134	0000H		CLOSEINCLUDE . . . PROCEDURE EXTERNAL(45) STACK=0000H
138	0000H		CLOSEPRINT . . . . PROCEDURE EXTERNAL(47) STACK=0000H
130	0000H		CLOSESOURCE. . . . PROCEDURE EXTERNAL(43) STACK=0000H
140	0000H		CLOSESYMBOL. . . . PROCEDURE EXTERNAL(48) STACK=0000H
1	0000H		CM2. . . . . PROCEDURE STACK=0000H
148	00AFH	100	CMBODY . . . . . BYTE ARRAY(100) 152 241 374 378 381
148	0114H	1	CMBODYFULL . . . . BYTE 208 220 385
148	0000H	1	CMBVAR . . . . . BYTE BASED(CMPT) 211 256 271 305 310
148	0006H	2	CMCOUNTER. . . . . WORD 204 205 216 217 241 384
148	00AEH	1	CMERROR. . . . . BYTE PUBLIC 207 219 261 276 314 321 347 360 411
148	0116H	1	CMLISTOVERFLOW . . BYTE PUBLIC 386 412
148	000CH	80	CMNAME . . . . . BYTE ARRAY(80) 230 237 377 380 391
148	005DH	1	CMNLG. . . . . BYTE 230 237 392
148	0000H	2	CMNPT. . . . . WORD 148 377
148	0000H	1	CMNVAR . . . . . BYTE BASED(CMNPT)
148	0000H	1	CMPMVAR. . . . . BYTE BASED(PMPT) 340 342
148	0004H	2	CMPT . . . . . WORD PUBLIC 148 211 212 223 224 256 257 271 272 304 305 309 310 378
148	0000H	2	CMWVAR . . . . . WORD BASED(CMPT) 223
3			CODE . . . . . LITERALLY
148	005CH	1	CODEMACROFOUND . . BYTE
8			CODEMACROHEAD. . . LITERALLY

COPYRIGHT © 1981, 1982
SPECIAL PURPOSE
P.O. BOX
PACIFIC GROVE, CA 93950
SER. #

```
 17                  COLON. . . . . .  LITERALLY       345
 17                  COMMA. . . . . .  LITERALLY       312  326
 54  0000H           COPY . . . . . .  PROCEDURE EXTERNAL(14) STACK=0000H      340  391
  2                  CR . . . . . .  LITERALLY
147  0000H     1     CSEGSPEC . . . .  BYTE EXTERNAL(54)
147  0000H     1     CSEGTYPE . . . .  BYTE EXTERNAL(52)
147  0000H     2     CSEGVALUE. . . .  WORD EXTERNAL(53)
 12                  CSOVER . . . . .  LITERALLY
147  0000H     2     CURDSEG. . . . .  WORD EXTERNAL(56)
 87  0000H     2     D. . . . . .  WORD PARAMETER        88
 81  0000H     2     D. . . . . .  WORD PARAMETER        82
 54  0000H     2     D. . . . . .  WORD PARAMETER        55
 84  0000H     2     D. . . . . .  WORD PARAMETER        85
 90  0000H     2     D. . . . . .  WORD PARAMETER        91
  2                  DCL. . . . . .  LITERALLY
 90  0000H           DECOUT . . . . .  PROCEDURE EXTERNAL(26) STACK=0000H
  3                  DELETEDSYMB. . .  LITERALLY
147  0001H     1     DESCR. . . . . .  BYTE MEMBER(TOKEN)       189  191  198
 33  0000H     2     DEST . . . . . .  WORD PARAMETER        34
142  0000H     1     DEV. . . . . .  BYTE PARAMETER       143
 66  0000H           DIGIT. . . . . .  PROCEDURE BYTE EXTERNAL(18) STACK=0000H
185  011AH     1     DISP . . . . . .  BYTE       188  190  192  193
  3                  DOUBLEDEFINED. .  LITERALLY
 16                  DOUBLERANGE. . .  LITERALLY       310
308  02DCH    53     DRANGE . . . . .  PROCEDURE STACK=0014H        327
 12                  DSOVER . . . . .  LITERALLY
147  0000H     1     DSPEC. . . . . .  BYTE EXTERNAL(55)
  4                  DWRD . . . . . .  LITERALLY       191  198
  2                  ELSEDO . . . . .  LITERALLY
105  0000H           EMPTYLINE. . . .  PROCEDURE BYTE EXTERNAL(32) STACK=0000H
  2                  ENDCASE. . . . .  LITERALLY
  2                  ENDDO. . . . . .  LITERALLY
  2                  ENDFOREVER . . .  LITERALLY
  2                  ENDIF. . . . . .  LITERALLY       179  238
  2                  ENDMODULE. . . .  LITERALLY
  2                  ENDOFFILE. . . .  LITERALLY
 19  0000H     2     ENDOFSYMBTAB . .  WORD EXTERNAL(1)
  2                  ENDPROC. . . . .  LITERALLY       323  414
  2                  ENDWHILE . . . .  LITERALLY
 36  0000H           ENTERATTRIBUTES.  PROCEDURE EXTERNAL(8) STACK=0000H
 81  0000H           EQUAL. . . . . .  PROCEDURE BYTE EXTERNAL(23) STACK=0000H       153
  3                  ERROR. . . . . .  LITERALLY
 12                  ESOVER . . . . .  LITERALLY
  2                  FALSE. . . . . .  LITERALLY       159  238  242  385  386  390
 60  0000H     2     FCBADR . . . . .  WORD PARAMETER        61
 60  0000H           FILEABORT. . . .  PROCEDURE EXTERNAL(16) STACK=0000H
  8                  FILEISTRUCTURE .  LITERALLY
  8                  FILEOSTRUCTURE .  LITERALLY
145  0000H           FILESETUP. . . .  PROCEDURE BYTE EXTERNAL(50) STACK=0000H
 63  0000H           FILL . . . . . .  PROCEDURE EXTERNAL(17) STACK=0000H       380  381  382
142  0000H     2     FILNAM . . . . .  WORD PARAMETER       143
142  0000H     2     FILTYP . . . . .  WORD PARAMETER       143
 39  0000H           FINDCODEMACRO. .  PROCEDURE BYTE EXTERNAL(9) STACK=0000H        230
 30  0000H           FINDSYMBOL . . .  PROCEDURE BYTE EXTERNAL(6) STACK=0000H
 17                  FIRST. . . . . .  LITERALLY       254  269  306  311
  2                  FOREVER. . . . .  LITERALLY
396  0479H    29     FORMALLISTROUT .  PROCEDURE PUBLIC STACK=0020H
```

Cop... 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. #:

```
  2                 FORMFEED . . . . .   LITERALLY
 18  0000H     2    FREEPT . . . . . .   WORD EXTERNAL(0)          240
 33  0000H          GETATTRIBUTES. . .   PROCEDURE EXTERNAL(7) STACK=0000H
148  0113H     1    GLOBALCHERROR. . .   BYTE PUBLIC           405
 84  0000H          HEX1OUT. . . . . .   PROCEDURE EXTERNAL(24) STACK=0000H
 87  0000H          HEX2OUT. . . . . .   PROCEDURE EXTERNAL(25) STACK=0000H
150  0117H     1    I. . . . . .         BYTE       151  153  155  157
171  0119H     1    I. . . . . .         BYTE       173  174  175  176  182
162  0118H     1    I. . . . . .         BYTE       164  165  166  168
  3                 IDENT. . . . . .     LITERALLY       389  397
142  0000H          IFILESETUP . . . .   PROCEDURE EXTERNAL(49) STACK=0000H
  8                 INDEXREGBIT. . . .   LITERALLY
  8                 INDEXREGCOUNT. . .   LITERALLY
118  0000H          ININCLUDEBYTE. . .   PROCEDURE BYTE EXTERNAL(37) STACK=0000H
  2                 INIT . . . . . .     LITERALLY
376  0400H    79    INITCMROUT . . . .   PROCEDURE PUBLIC STACK=000AH
116  0000H          INSOURCEBYTE . . .   PROCEDURE BYTE EXTERNAL(36) STACK=0000H
  8                 IREGBIT. . . . . .   LITERALLY
  8                 IREGCOUNT. . . . .   LITERALLY
170  0004H     1    L. . . . . .         BYTE PARAMETER AUTOMATIC       171  174  176
161  0004H     1    L. . . . . .         BYTE PARAMETER AUTOMATIC       162  165
335  0004H     1    L. . . . . .         BYTE PARAMETER AUTOMATIC       336  337
331  0004H     1    L. . . . . .         BYTE PARAMETER AUTOMATIC       332  333
245  0004H     1    L. . . . . .         BYTE PARAMETER AUTOMATIC       246  247
  3                 LAB. . . . . .       LITERALLY
                    LAST . . . . . .     BUILTIN        164  173
  7                 LEFTBRACKET. . . .   LITERALLY
 17                 LEFTPAR. . . . . .   LITERALLY      372
170  009AH    76    LEGALMODLETTER . .   PROCEDURE BYTE STACK=0004H          364
149  0000H   113    LEGALPARAMETER . .   PROCEDURE BYTE PUBLIC STACK=0012H
184  00E3H    55    LEGALREGISTER. . .   PROCEDURE BYTE STACK=0002H          274
195  011DH    27    LEGALSEGREG. . . .   PROCEDURE BYTE PUBLIC STACK=0002H
161  0071H    41    LEGALSPECLETTER. .   PROCEDURE BYTE STACK=0004H          350
                    LENGTH . . . . . .   BUILTIN        380  381  382
147  0000H     2    LENGTH . . . . . .   WORD MEMBER(OPERANDS)
 69  0000H          LETTER . . . . . .   PROCEDURE BYTE EXTERNAL(19) STACK=0000H
  2                 LF . . . . . .       LITERALLY
149  0008H     1    LG . . . . . .       BYTE PARAMETER AUTOMATIC       150  153
 45  0000H     1    LG . . . . . .       BYTE PARAMETER          46
 42  0000H     1    LG . . . . . .       BYTE PARAMETER          43
 39  0000H     1    LG . . . . . .       BYTE PARAMETER          40
 30  0000H     1    LG . . . . . .       BYTE PARAMETER          31
 27  0000H     1    LG . . . . . .       BYTE PARAMETER          28
 24  0000H     1    LG . . . . . .       BYTE PARAMETER          25
227  0003H     2    LISTPTR. . . . . .   WORD       227  229  232  233  240
  2                 LIT. . . . . .       LITERALLY        2    3    4    5    6    7    8    9   10   11   12
                                           13   14   15   16   17
 27  0000H     2    MACDEFPT . . . . .   WORD PARAMETER          28
  9                 MDBF . . . . . .     LITERALLY
  9                 MDBIT. . . . . .     LITERALLY
  9                 MDBN . . . . . .     LITERALLY
  9                 MDDF . . . . . .     LITERALLY
  9                 MDWF . . . . . .     LITERALLY
  9                 MDWN . . . . . .     LITERALLY
  9                 MENDBIT. . . . . .   LITERALLY
  9                 MENDM. . . . . .     LITERALLY
  9                 MFORMALBITS. . . .   LITERALLY
```

```
  9                MMODRM1. . . . . .  LITERALLY
  9                MMODRM2. . . . . .  LITERALLY
  9                MNOSEGFIX. . . . .  LITERALLY
  9                MNUMBERBITS. . . .  LITERALLY
 11                MODB . . . . . .    LITERALLY       365
 11                MODD . . . . . .    LITERALLY       367
 15                MODLETTERBIT . . .  LITERALLY
 15                MODLETTERCOUNT . .  LITERALLY
335  031EH    13  MODLETTERROUT. . .  PROCEDURE STACK=000AH          365  366  367  368  369
 11                MODSB. . . . . .    LITERALLY       368
 11                MODW . . . . . .    LITERALLY       366
  9                MRELB. . . . . .    LITERALLY
  9                MRELW. . . . . .    LITERALLY
  9                MSEGFIX. . . . . .  LITERALLY
 90  0000H     2  N. . . . . .        WORD PARAMETER        91
 87  0000H     2  N. . . . . .        WORD PARAMETER        88
 84  0000H     1  N. . . . . .        BYTE PARAMETER        85
 81  0000H     1  N. . . . . .        BYTE PARAMETER        82
 63  0000H     1  N. . . . . .        BYTE PARAMETER        64
 54  0000H     1  N. . . . . .        BYTE PARAMETER        55
 51  0000H     2  N. . . . . .        WORD PARAMETER        52
338  044FH    42  NAMEROUT . . . . .  PROCEDURE BYTE PUBLIC STACK=000AH
  3                NEGLECTED. . . . .  LITERALLY
 42  0000H         NEWCMBODY. . . . .  PROCEDURE BYTE EXTERNAL(10) STACK=0000H      241
 45  0000H         NEWCMNAME. . . . .  PROCEDURE BYTE EXTERNAL(11) STACK=0000H      237
 27  0000H         NEWMACRO . . . . .  PROCEDURE BYTE EXTERNAL(5) STACK=0000H
 24  0000H         NEWSYMBOL. . . . .  PROCEDURE BYTE EXTERNAL(4) STACK=0000H
227  0000H     2  NEXT . . . . . .    WORD BASED(LISTPTR)       232  233  240
147  0000H     1  NEXTCH . . . . . .  BYTE EXTERNAL(58)        326
  4                NIL. . . . . .      LITERALLY
150  0000H     1  NO . . . . . .      BYTE BASED(PTR2)          152
 11                NOMOD. . . . . .    LITERALLY       369
147  0000H     1  NOOPER . . . . . .  BYTE EXTERNAL(61)
  8                NOOVERCOUNT. . . .  LITERALLY
  8                NOOVERRIDEBIT. . .  LITERALLY
 13                NOPARAND . . . . .  LITERALLY
 16                NORANGE. . . . . .  LITERALLY
  3                NUMBER . . . . . .  LITERALLY       252
 16                NUMBERRANGE. . . .  LITERALLY       256
  7                OAND . . . . . .    LITERALLY
  7                OEQ. . . . . .      LITERALLY
147  0006H     2  OFFSET . . . . . .  WORD MEMBER(OPERANDS)
  7                OGE. . . . . .      LITERALLY
  7                OGT. . . . . .      LITERALLY
  7                OLAST. . . . . .    LITERALLY
  7                OLE. . . . . .      LITERALLY
  7                OLENGTH. . . . . .  LITERALLY
  7                OLT. . . . . .      LITERALLY
  7                OMOD . . . . . .    LITERALLY
  7                ONE. . . . . .      LITERALLY
  7                ONOT . . . . . .    LITERALLY
  7                OOFFSET. . . . . .  LITERALLY
  7                OOR. . . . . .      LITERALLY
124  0000H         OPENHEX. . . . . .  PROCEDURE EXTERNAL(40) STACK=0000H
122  0000H         OPENINCLUDE. . . .  PROCEDURE EXTERNAL(39) STACK=0000H
123  0000H         OPENPRINT. . . . .  PROCEDURE EXTERNAL(41) STACK=0000H
120  0000H         OPENSOURCE . . . .  PROCEDURE EXTERNAL(38) STACK=0000H
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
128  0000H        OPENSYMBOL . . . .   PROCEDURE EXTERNAL(42) STACK=0000H
147  0000H   36   OPERANDS . . . . .   STRUCTURE ARRAY(4) EXTERNAL(62)
  8               OPERANDSTRUC . . .   LITERALLY       147
  3               OPERATOR . . . . .   LITERALLY
  7               OPTR . . . . . .     LITERALLY
  7               OSEG . . . . . .     LITERALLY
  7               OSHL . . . . . .     LITERALLY
  7               OSHORT . . . . .     LITERALLY
  7               OSHR . . . . . .     LITERALLY
  7               OTYPE. . . . . .     LITERALLY
107  0000H        OUTHEXBYTE . . . .   PROCEDURE EXTERNAL(33) STACK=0000H
110  0000H        OUTPRINTBYTE . . .   PROCEDURE EXTERNAL(34) STACK=0000H
113  0000H        OUTSYMBOLBYTE. . .   PROCEDURE EXTERNAL(35) STACK=0000H
 57  0000H        OUTTEXT. . . . . .   PROCEDURE EXTERNAL(15) STACK=0000H
  7               OXOR . . . . . .     LITERALLY
339  032BH  213   PARDESCRROUT . . .   PROCEDURE STACK=001CH          398
148  005EH   80   PARNAME. . . . . .   BYTE ARRAY(80)       153  155  379  382
  6               PCODEMACRO . . . .   LITERALLY
  6               PCSEG. . . . . .     LITERALLY
  6               PDB. . . . . .       LITERALLY
  6               PDBIT. . . . . .     LITERALLY
  6               PDD. . . . . .       LITERALLY
  6               PDSEG. . . . . .     LITERALLY
  6               PDW. . . . . .       LITERALLY
  6               PEJECT . . . . . .   LITERALLY
  6               PEND . . . . . .     LITERALLY
  6               PENDIF . . . . . .   LITERALLY
  6               PENDM. . . . . .     LITERALLY
  6               PEQU . . . . . .     LITERALLY
  6               PESEG. . . . . .     LITERALLY
  6               PIF: . . . . . .     LITERALLY
  6               PIFLIST. . . . . .   LITERALLY
  6               PINCLUDE . . . . .   LITERALLY
  6               PLIST. . . . . .     LITERALLY
  6               PMODRM . . . . . .   LITERALLY
148  0002H    2   PNPT . . . . . .     WORD       148  340  341  342  343  379
  6               PNOIFLIST. . . . .   LITERALLY
  6               PNOLIST. . . . . .   LITERALLY
  6               PNOSEGFIX. . . . .   LITERALLY
  6               PORG . . . . . .     LITERALLY
  6               PPAGESIZE. . . . .   LITERALLY
  6               PPAGEWIDTH . . . .   LITERALLY
  6               PRB. . . . . .       LITERALLY
 13               PREFIXAND. . . . .   LITERALLY
 14               PREFIXON . . . . .   LITERALLY
  6               PRELB. . . . . .     LITERALLY
  6               PRELW. . . . . .     LITERALLY
  2               PROC . . . . . .     LITERALLY        20   22   24   27   30   33   36   39   42   45   48
                                         51   72   75   84   87   90   93   95   97   99  102  105  107  110
                                        113  116  118  120  122  124  126  128  130  132  134  136  138  140
                                        142  145  149  161  170  184  195  202  214  226  245  249  250  303
                                        308  331  335  339  376  388  396  404
  6               PRS. . . . . .       LITERALLY
  6               PRW. . . . . .       LITERALLY
  6               PSEGFIX. . . . . .   LITERALLY
  3               PSEUDO . . . . . .   LITERALLY
  6               PSIMFORM . . . . .   LITERALLY
```

COPYRIGHT © 1976, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

```
  8              PSSEG. . . . . .   LITERALLY
 63  0000H    2  PT . . . . . .     WORD PARAMETER       64
  6              PTITLE . . . . .   LITERALLY
228  000AH    2  PTR. . . . . .     WORD       229  230  237
149  0006H    2  PTR. . . . . .     WORD PARAMETER AUTOMATIC      150  153
 42  0000H    2  PTR. . . . . .     WORD PARAMETER       43
149  0004H    2  PTR2 . . . . . .   WORD PARAMETER AUTOMATIC      150  152
202  0138H   44  PUTB . . . . . .   PROCEDURE PUBLIC STACK=0004H       247  265  333  337
250  021DH  168  PUTRANGE . . . . .  PROCEDURE STACK=0010H         306  311  318
214  0164H   47  PUTW . . . . . .   PROCEDURE PUBLIC STACK=0004H
  5              RAH. . . . . .     LITERALLY       291
  5              RAL. . . . . .     LITERALLY       287
 16              RANGEAND . . . . .  LITERALLY
249  0203H   26  RANGEROUT. . . . .  PROCEDURE STACK=0018H         373
 15              RANGESPECBIT . . .  LITERALLY
 15              RANGESPECCOUNT . .  LITERALLY
 16              RANGETYPEAND . . .  LITERALLY
  5              RAX. . . . . .     LITERALLY       279
  5              RBH. . . . . .     LITERALLY       294
  5              RBL. . . . . .     LITERALLY       290
  5              RBP. . . . . .     LITERALLY       284
  5              RBX. . . . . .     LITERALLY       282
  5              RCH. . . . . .     LITERALLY       292
  5              RCL. . . . . .     LITERALLY       288
  5              RCS. . . . . .     LITERALLY       296
  5              RCX. . . . . .     LITERALLY       280
  5              RDH. . . . . .     LITERALLY       293
  5              RDI. . . . . .     LITERALLY       286
  5              RDL. . . . . .     LITERALLY       289
  5              RDS. . . . . .     LITERALLY       298
  5              RDX. . . . . .     LITERALLY       281
  2              REENT. . . . . .   LITERALLY
  3              REG. . . . . .     LITERALLY       186  196
 16              REGISTERRANGE. . .  LITERALLY       271
245  01F6H   13  REGISTERROUT . . .  PROCEDURE STACK=000AH         279  280  281  282  283  284  285  286
                                     287  288  289  290  291  292  293  294  295  296  297  298
  5              RES. . . . . .     LITERALLY       275
 39  0000H    2  RESULT . . . . . .  WORD PARAMETER       40
 30  0000H    2  RESULT . . . . . .  WORD PARAMETER       31
 24  0000H    2  RESULT . . . . . .  WORD PARAMETER       25
 45  0000H    2  RETURNPTR. . . . .  WORD PARAMETER       46
132  0000H       REWINDSOURCE . . .  PROCEDURE EXTERNAL(44) STACK=0000H
  7              RIGHTBRACKET . . .  LITERALLY
 17              RIGHTPAR . . . . .  LITERALLY       319
  5              RSI. . . . . .     LITERALLY       285
  5              RSP. . . . . .     LITERALLY       283
  5              RSS. . . . . .     LITERALLY       297
 81  0000H    2  S. . . . . .       WORD PARAMETER       82
 54  0000H    2  S. . . . . .       WORD PARAMETER       55
 95  0000H       SCAN . . . . . .   PROCEDURE EXTERNAL(28) STACK=0000H      301  317  325  329  344  349
                                     371  393  401
 93  0000H       SCANINIT . . . . .  PROCEDURE EXTERNAL(27) STACK=0000H
 17              SECOND . . . . . .  LITERALLY       318
  8              SEGMBIT. . . . . .  LITERALLY
  8              SEGMCOUNT. . . . .  LITERALLY
147  0004H    2  SEGMENT. . . . . .  WORD MEMBER(OPERANDS)
  8              SEGTYPEBIT . . . .  LITERALLY
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
  8                SEGTYPECOUNT . . .  LITERALLY
147  0003H     1    SFLAG. . . . . .    BYTE MEMBER(OPERANDS)
 16                SINGLERANGE. . . .  LITERALLY       305
 97  0000H         SKIPRESTOFLINE . .  PROCEDURE EXTERNAL(29) STACK=0000H
102  0000H         SKIPUNTIL. . . . .  PROCEDURE BYTE EXTERNAL(31) STACK=0000H
 36  0000H     2   SOURCE . . . . . .  WORD PARAMETER        37
  2                SPACE. . . . . .    LITERALLY
  3                SPEC . . . . . .    LITERALLY
 10                SPECA. . . . . .    LITERALLY       351
 10                SPECC. . . . . .    LITERALLY       352
 10                SPECD. . . . . .    LITERALLY       353
 10                SPECE. . . . . .    LITERALLY       354
 99  0000H         SPECIALTOKEN . . .  PROCEDURE BYTE EXTERNAL(30) STACK=0000H
331  0311H    13   SPECLETTERROUT . .  PROCEDURE STACK=000AH          351  352  353  354  355  356  357  358
 10                SPECM. . . . . .    LITERALLY       355
 10                SPECR. . . . . .    LITERALLY       356
 10                SPECS. . . . . .    LITERALLY       357
 10                SPECX. . . . . .    LITERALLY       358
303  02C5H    23   SRANGE . . . . . .  PROCEDURE STACK=0014H          328
 12                SSOVER . . . . . .  LITERALLY
 39  0000H     2   STRADR . . . . . .  WORD PARAMETER        40
 30  0000H     2   STRADR . . . . . .  WORD PARAMETER        31
 27  0000H     2   STRADR . . . . . .  WORD PARAMETER        28
 24  0000H     2   STRADR . . . . . .  WORD PARAMETER        25
  3                STRING . . . . . .  LITERALLY
  2                STRUC. . . . . .    LITERALLY       147
147  0002H     1   STYPE. . . . . .    BYTE MEMBER(OPERANDS)
 36  0000H     2   SYMBADR. . . . . .  WORD PARAMETER        37
 33  0000H     2   SYMBADR. . . . . .  WORD PARAMETER        34
 22  0000H         SYMBINIT . . . . .  PROCEDURE EXTERNAL(3) STACK=0000H
  3                SYMBOL . . . . . .  LITERALLY
  8                SYMBOLHEAD . . . .  LITERALLY
  3                SYMBOLSTRUC. . . .  LITERALLY       147
 20  0000H         SYMBTERMINATE. . .  PROCEDURE EXTERNAL(2) STACK=0000H
 57  0000H     2   T. . . . . .        WORD PARAMETER        58
  2                TAB. . . . . .      LITERALLY
172  0003H     4   TABLE. . . . . .    BYTE ARRAY(4) DATA        173  174  176
163  0000H     8   TABLE. . . . . .    BYTE ARRAY(8) DATA        164  165
404  0496H    35   TERMINATECMROUT. .  PROCEDURE PUBLIC STACK=000EH
 60  0000H     2   TEXTADR. . . . . .  WORD PARAMETER        61
  2                THENDO . . . . . .  LITERALLY       176  205  217  230  252  254  259  269  312  319  345
                                        405  409
250  0004H     1   TIME . . . . . .    BYTE PARAMETER AUTOMATIC       251  254  269
102  0000H     1   TOK. . . . . .      BYTE PARAMETER       103
 99  0000H     1   TOK. . . . . .      BYTE PARAMETER       100
147  0000H     4   TOKEN. . . . . .    STRUCTURE EXTERNAL(57)         186  189  191  193  196  198  200  252
                                        259  265  389  397
  2                TRUE . . . . . .    LITERALLY       154  207  208  219  220  243  261  276  314  321  347
                                        360  394  411  412
147  0000H     1   TYPE . . . . . .    BYTE MEMBER(TOKEN)        186  196  252  389  397
  8                TYPEBIT. . . . . .  LITERALLY
 48  0000H         TYPECALC . . . . .  PROCEDURE BYTE EXTERNAL(12) STACK=0000H
  8                TYPECOUNT. . . . .  LITERALLY
  3                UDEFSYMB . . . . .  LITERALLY
226  0193H    99   UPDATECMLISTS. . .  PROCEDURE BYTE STACK=000AH          409
 78  0000H         UPPER. . . . . .    PROCEDURE BYTE EXTERNAL(22) STACK=0000H
 48  0000H     2   VAL. . . . . .      WORD PARAMETER        49
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
  147  0002H     2  VALUE. . . . . .   WORD MEMBER(TOKEN)          193  200  259  265
    3               VARIABLE . . . . .  LITERALLY
  214  0004H     2  W. . . . . .       WORD PARAMETER AUTOMATIC    215  223
    4               WRD. . . . . .     LITERALLY
   51  0000H        WRDTEST. . . . .   PROCEDURE BYTE EXTERNAL(13) STACK=0000H
```

MODULE INFORMATION:

```
CODE AREA SIZE     = 04B9H   1209D
CONSTANT AREA SIZE = 000CH     12D
VARIABLE AREA SIZE = 011BH    283D
MAXIMUM STACK SIZE = 0020H     32D
936 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____